

2023 中国软件供应链 安全分析报告

THE REPORT

发布机构：

奇安信代码安全实验室



目 录

一、概述	1
1、软件供应链安全攻击事件依然高发	1
2、开源软件安全是软件供应链安全的重中之重	3
二、国内企业自主开发源代码安全状况	5
1、编程语言分布情况	5
2、典型安全缺陷检出情况	6
三、开源软件生态发展与安全状况	7
1、开源软件生态发展状况分析	7
2、开源软件源代码安全状况分析	9
(1) 编程语言分布情况	10
(2) 典型安全缺陷检出情况	10
(3) 安全问题修复确认情况	11
3、开源软件公开报告漏洞状况分析	11
(1) 大型开源项目漏洞总数及年度增长 TOP20	12
(2) 主流开源软件包生态系统漏洞总数及年度增长 TOP20	14
4、开源软件活跃度状况分析	15

(1) 超 7 成开源软件项目处于不活跃状态	16
(2) 超 2.2 万个开源软件一年内更新发布超过 100 个版本	16
5、关键基础开源软件分析	17
(1) 主流开源生态关键基础开源软件 TOP50	17
(2) 从未公开披露过漏洞的关键基础开源软件占比进一步升高	20
(3) 关键基础开源软件的整体运维风险仍处于较高水平	20
四、国内企业软件开发中开源软件应用状况	21
1、开源软件总体使用情况分析	21
(1) 平均每个软件项目使用 155 个开源软件	21
(2) 流行开源软件被超 4 成的软件项目使用	22
2、开源软件漏洞风险分析	22
(1) 近 8 成软件项目存在容易利用的开源软件漏洞	22
(2) 平均每个软件项目存在 110 个已知开源软件漏洞	23
(3) 影响最广的开源软件漏洞存在于超 4 成的软件项目中	24
(4) 20 多年前的开源软件漏洞仍然存在于多个软件项目中	25
3、开源软件许可协议风险分析	26
(1) 最流行的开源许可协议在超半数的项目中使用	26
(2) 1/6 的项目使用了含有超、高危许可协议的开源软件	26

4、开源软件运维风险分析	28
(1) 近 30 年前的老旧开源软件版本仍在被使用	28
(2) 开源软件各版本使用依然混乱	29
5、不同行业软件项目开源使用风险分析	29
(1) 各行业软件项目分布情况	29
(2) 各行业软件项目使用开源软件情况	30
(3) 各行业软件项目含已知开源软件漏洞情况	31
五、典型软件供应链安全风险实例分析	32
1、某主流企业级无线路由器供应链攻击实例分析	32
2、ZCS 协同办公系统供应链攻击实例分析	33
3、多款国产操作系统供应链攻击实例分析	35
4、某国产 CMS 系统供应链攻击实例分析	37
六、总结及建议	39
附录：奇安信代码安全实验室简介	42

一、概述

过去的一年，各界对软件供应链安全的关注度依然高涨，相关安全攻击事件也持续增加。为此，奇安信代码安全实验室在前两期《中国软件供应链安全分析报告》(https://www.qianxin.com/threat/reportdetail?report_id=161)的基础上，推出本分析报告。

作为该系列年度分析报告的第三期，本报告继续针对国内企业自主开发的源代码、开源软件生态、国内企业软件开发中开源软件应用等的安全状况，以及典型应用系统供应链安全风险实例进行深入分析，并总结趋势和变化。相比于去年的报告，本报告有以下新增之处：在开源软件生态发展与安全部分新增了开源项目维护者对他人提交安全问题的修复确认情况；在企业软件开发中的开源软件应用部分新增了对不同行业软件项目开源使用风险的分析，对开源许可协议的风险分析分别统计了超危和高危开源许可协议的使用情况；在典型软件供应链安全风险实例部分，通过多个新的实例再次验证了因软件供应链的复杂性，开源软件的“老漏洞”发挥“0day 漏洞”攻击作用的状况。感兴趣的读者阅读时可重点关注上述变化之处。

1、软件供应链安全攻击事件依然高发

在过去的一年中，针对软件供应链的安全攻击事件持续增加，造成的危害也愈发严重。

2022 年 7 月，攻击者通过在 NPM 上发布包时绕过双因子认证

(2FA)，创建了 1000 多个用户账号，攻击者利用这些账号自动投放 1283 个包含挖矿脚本 eazyminer 的恶意模块，利用数据库、Web 等所在服务器的机器闲置资源进行挖矿，如果开发者安装了这些包，则会在被调用时挖掘门罗币。

2022 年 11 月，美国 FBI 和 CISA 发布联合公告指出，未具名的伊朗组织利用 Log4Shell 漏洞入侵了美联邦民事行政部门 FCEB 下属机构的一台未修复的 VMware Horizon 服务器，并部署了挖矿恶意软件 XMRig。FBI 和 CISA 提到，所有尚未修复 VMware 系统中 Log4Shell 漏洞的机构都可能遭受此类攻击。

2022 年 12 月，安全研究人员发现了 GitHub Actions 的一个严重漏洞，该漏洞可造成恶意软件植入攻击，进而影响使用 GitHub Actions 的软件项目，如下游软件项目以恶意代码编译更新等。GitHub 证实了该问题的存在，并颁发了赏金，Rust 修复了易受攻击的管道。

2023 年 2 月，半导体行业技术巨头 Applied Materials 称：其一家主要的供应商因遭受勒索软件攻击而被迫中断生产，这将影响其第二季度的交付。该攻击事件及正在进行的其他供应链挑战将使其第二季度的成本增加约 2.5 亿美元。

2023 年 3 月，安全研究人员发现，互联网语音协议交换机 (VoIP IPBX) 软件开发厂商 3CX 的 VoIP 桌面客户端在通过 Git 进行构建时，注入了因供应链攻击而引入的恶意代码，并使用合法的 3CX Ltd 证书进行了数字签名，其中一个恶意 DLL 是利用了一个存在近 10 年的 Windows 签名验证漏洞 (CVE-2013-3900) 通过签名的。当用户安装应

用时，会加载恶意 DLL，进而收集系统信息、窃取数据和凭据等，造成敏感数据泄露。3CX Phone System 被全球超 60 万家企业使用，每天用户超过 1200 万。

2023 年 5 月，网络安全研究员对适用于 Linux 和 Unix 平台的热门开源互联网路由协议套件 FRRouting 中网关协议 BGP 的 7 种不同实现软件进行了分析，发现存在三个漏洞，它们的 CVSS 评分均为 6.5，可被用于在易受攻击的 BGP 对等体上实现拒绝服务条件。目前 FRRouting 用于多家厂商如 NVIDIA Cumulus、DENT 和 SONiC 中。

2、开源软件安全是软件供应链安全的重中之重

奇安信代码安全实验室通过数据分析发现，与前两年相比，开源软件自身的安全状况持续下滑，国内企业软件开发中因使用开源软件而引入安全风险的状况更加糟糕。此外，开源项目维护人员对安全问题的修复积极性较低。开源软件供应链安全风险管控依然值得持续关注，需要更大的投入。

1) 开源生态发展依然迅猛，开源软件自身安全状况持续下滑

过去一年，主流开源软件包生态系统中开源项目总量增长了 25.1%，开源生态发展依然迅猛。与此同时，开源软件漏洞数量也持续增长，2022 年新增的开源软件漏洞为 7682 个；三年来不活跃的开源项目占比从 61.6%、69.9% 逐渐升高至 72.1%。根据“奇安信开源项目检测计划”的实测数据显示，三年来开源软件的总体缺陷密度和高危缺陷密度呈现出逐年上升的趋势，均处于较高水平；十类典型缺陷

的总体检出率为 72.3%，与去年 73.5%相当，远高于前年的 56.3%。总体来看，开源软件自身的安全问题愈发严峻。

2) 开源项目维护者对安全问题的重视度和修复积极性较低

2022 年，“奇安信开源项目检测计划”共向各被测开源项目的维护者反馈 1484 个安全问题，仅有 547 个得到确认并修复，其他 937 个反馈不修复、未反馈，或无人处理，安全问题的修复率仅为 36.9%。另据统计发现，一个安全问题从提交到维护人员反馈确认并修复，时间较长的可长达一年甚至更久。

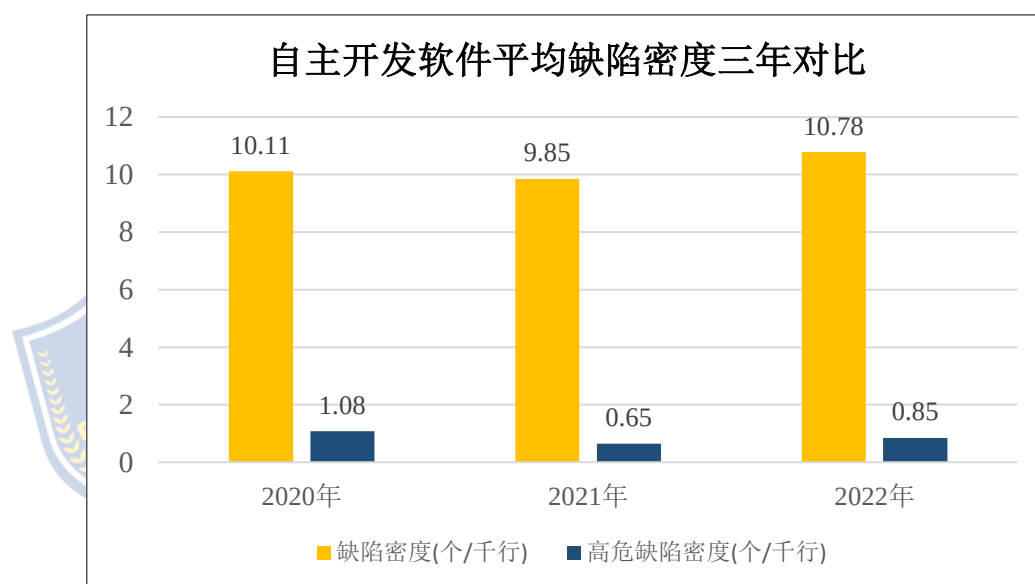
3) 国内企业因使用开源软件而引入安全风险的状况更加糟糕

2022 年，奇安信代码安全实验室对 2631 个国内企业软件项目中使用开源软件的情况进行分析发现：平均每个项目使用 155 个开源软件，远高于之前的 126 和 127 个；存在已知开源软件漏洞的项目占比达 91.6%，平均每个项目存在 110 个已知开源软件漏洞，项目中最古老的开源软件漏洞的发现时间可追溯至 21 年前；1/6 的项目中使用了含有超危或高危许可协议的开源软件；近 30 年前的老旧开源软件版本仍然在使用，同一开源软件各版本的使用依然混乱。整体的安全风险状况与前两年相比更加糟糕，风险水平仍处于较高状态。

另一方面，我们发现 2022 年许多机构和企业更加关注开源软件供应链安全，一些机构和企业基于规范的流程，在开源安全治理工具的辅助下开展相关工作。但从目前国内企业软件开发中使用开源软件的安全风险状况来看，这些经验、方法和工具还需要进一步的持续推广和应用。

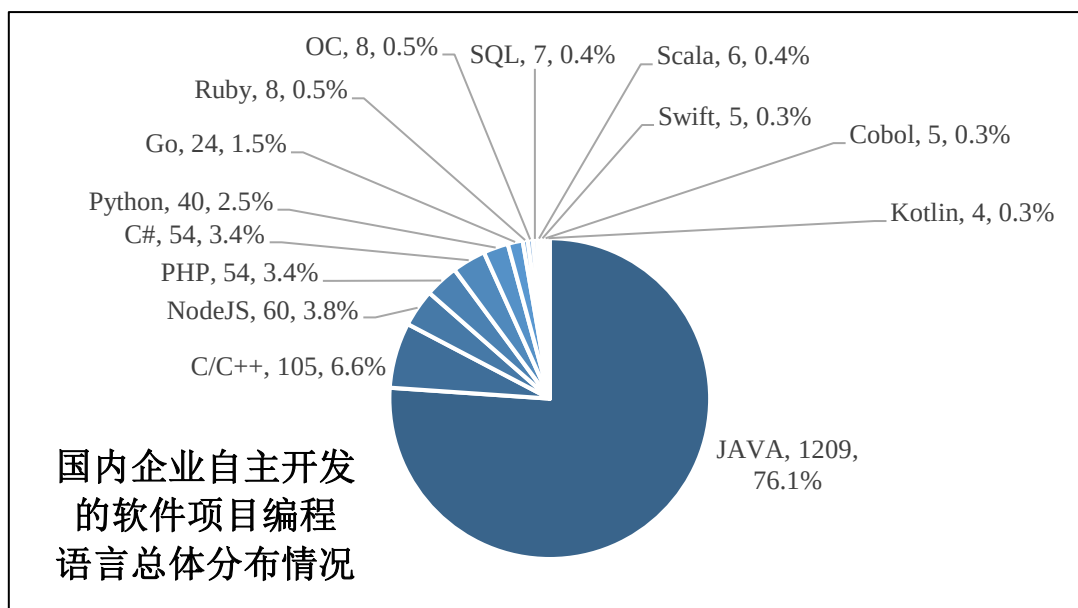
二、国内企业自主开发源代码安全状况

软件源代码安全是软件供应链安全的基础。2022 年全年，奇安信代码安全实验室对 1589 个国内企业自主开发的软件项目源代码进行了安全缺陷检测，检测的代码总量为 347814428 行，共发现安全缺陷 3751450 个，其中高危缺陷 300459 个，整体缺陷密度为 10.78 个/千行，高危缺陷密度为 0.85 个/千行。与前两年相比，整体缺陷密度略微升高，高危缺陷密度较去年有所反弹，但低于前年水平。



1、编程语言分布情况

在 1589 个国内企业自主开发的软件项目中，共使用了 14 种编程语言，使用数量排名前 3 的为 Java、C/C++、NodeJS，对应的软件项目数量分别为 1209 个、105 个和 60 个，其中 Java 语言项目占比达 76.1%。可以看出，国内企业在进行软件开发时，Java 语言仍然是首选，且比例更大。编程语言的分布情况如下图所示。



2、典型安全缺陷检出情况

分析 1589 个软件项目的源代码缺陷检测结果发现，注入、密码管理、日志伪造、跨站脚本、NULL 引用、配置管理、输入验证、资源管理、路径遍历、API 误用等十类典型安全缺陷的总体检出率（检出率指含有某类缺陷的软件项目数占软件项目总数的比例）为 74.1%，高于去年报告的 59.9%，与前年的 77.8% 相当。每类典型缺陷近三年的检出率及排名情况如下表所示。

排名	2022 检出率及排名	2021 检出率及排名	2020 检出率及排名
1	输入验证 (50.5%)	输入验证 (41.8%)	输入验证 (50.8%)
2	跨站脚本 (44.2%)	跨站脚本 (35.1%)	路径遍历 (39.6%)
3	资源管理 (43.3%)	API 误用 (31.5%)	跨站脚本 (39.5%)
4	NULL 引用 (43.1%)	NULL 引用 (30.2%)	注入 (37.3%)
5	注入 (40.1%)	资源管理 (29.9%)	NULL 引用 (31.8%)
6	路径遍历 (40.0%)	路径遍历 (28.4%)	资源管理 (31.6%)
7	密码管理 (30.3%)	注入 (26.3%)	密码管理 (31.0%)
8	API 误用 (30.1%)	密码管理 (22.8%)	API 误用 (28.7%)

9	配置管理 (24.8%)	配置管理 (18.2%)	配置管理 (28.0%)
10	日志伪造 (17.6%)	日志伪造 (12.5%)	日志伪造 (18.2%)

从表中可以看出，三年来，输入验证、跨站脚本、NULL 引用是检出率较高的缺陷类型，配置管理和日志伪造类缺陷检出率较低。除 API 误用类缺陷的检出率略有下降外，其他类型缺陷的检出率均比去年有不同程度的升高，又普遍回到或超过了前年的水平。国内企业在自主开发软件代码的安全质量方面不能掉以轻心。

三、开源软件生态发展与安全状况

开源软件在现代软件开发中的基础作用已得到普遍认可。本年度报告依然从开源软件生态发展状况、开源软件源代码安全状况、开源软件公开报告漏洞状况、开源软件活跃度状况、关键基础开源软件分析等五个方面对 2022 年开源软件生态发展与安全状况进行综合分析。

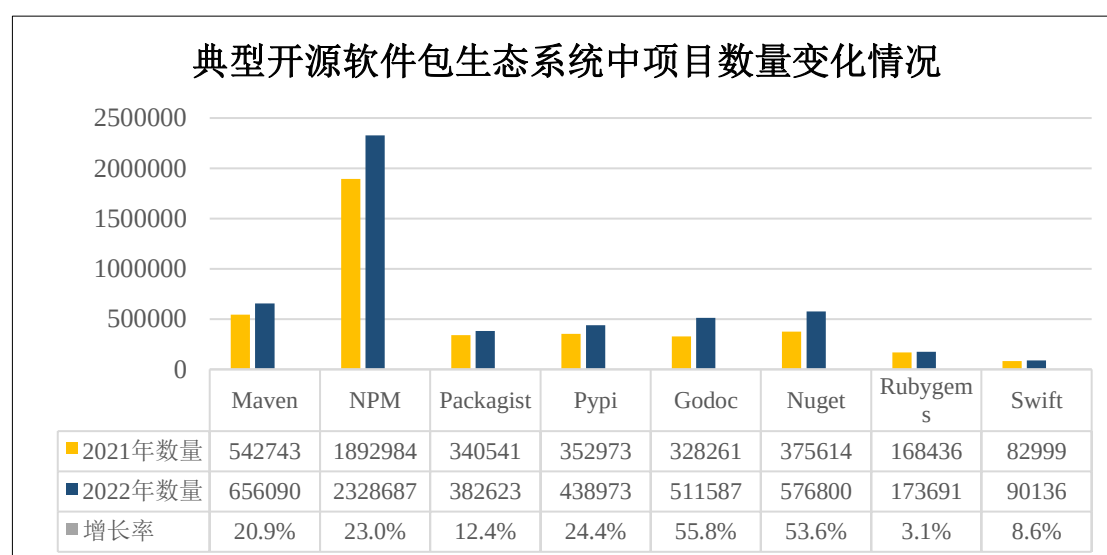
1、开源软件生态发展状况分析

根据奇安信代码安全实验室的监测和统计，2021 年底和 2022 年底，主流开源软件包生态系统中开源项目总量分别为 4395386 个和 5499977，一年间增长了 25.1%；截至 2022 年底，主流开源软件包生态系统中平均每个开源项目有 12.6 个版本，较前两年报告的 11.8 个和 10.2 个呈持续增长的趋势。可以看出，2022 年开源软件生态依然繁荣。

对 Maven、NPM、Packagist、Pypi、Godoc、Nuget、Rubygems、

Swift 等八个典型开源软件包生态系统的具体分析如下：

NPM 包生态项目数量依然最多，Godoc 包生态增速依然最快。这一状况与前两年报告保持一致。八个典型的开源软件包生态系统中开源项目数量和增长率情况如下图所示，其中开源项目数量最多的是 NPM 包生态系统，截至 2022 年底，其开源项目数量达到了 2328687 个，远高于其他开源包生态；开源项目数量增速最快的是 Godoc 包生态系统，2022 年一年间的项目总量增速达到了 55.8%，Nuget 的增长也很快，达 53.6%。

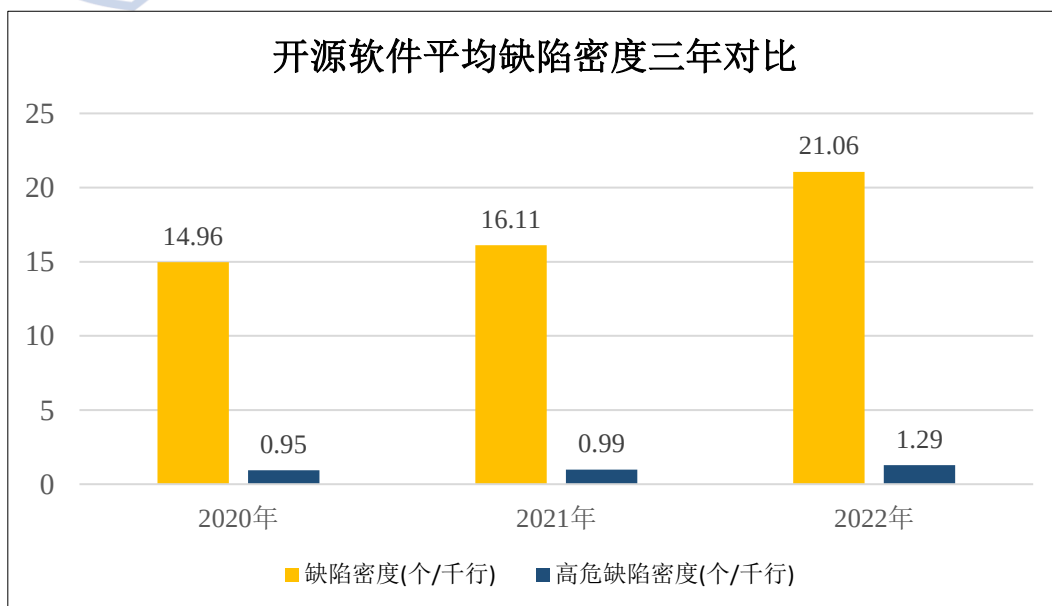


Maven 和 NPM 包生态系统的开源项目开发者仍然是“最勤奋”的，开源项目的平均版本数超过 13 个。截至 2022 年底，八个典型的开源软件包生态系统的开源项目数量和版本数量如下表所示。其中，Maven 包生态系统平均每个开源项目有高达 21.9 个版本，也是平均版本数增长最快的包生态系统；其他增长较快的还有 Packagist 和 Godoc；Nuget 是开源项目平均版本数唯一降低的包生态系统，从去年的 12.7 降为今年的 11.0。

序号	包生态系统	2022 年项目数	2022 年版本数	平均版本数
1	Maven	656090	14348900	21.9
2	NPM	2328687	31215362	13.4
3	Packagist	382623	4551007	11.9
4	Pypi	438973	4236546	9.7
5	Godoc	511587	4817892	9.4
6	Nuget	576800	6344100	11.0
7	Rubygems	173691	1290368	7.4
8	Swift	90136	604816	6.7

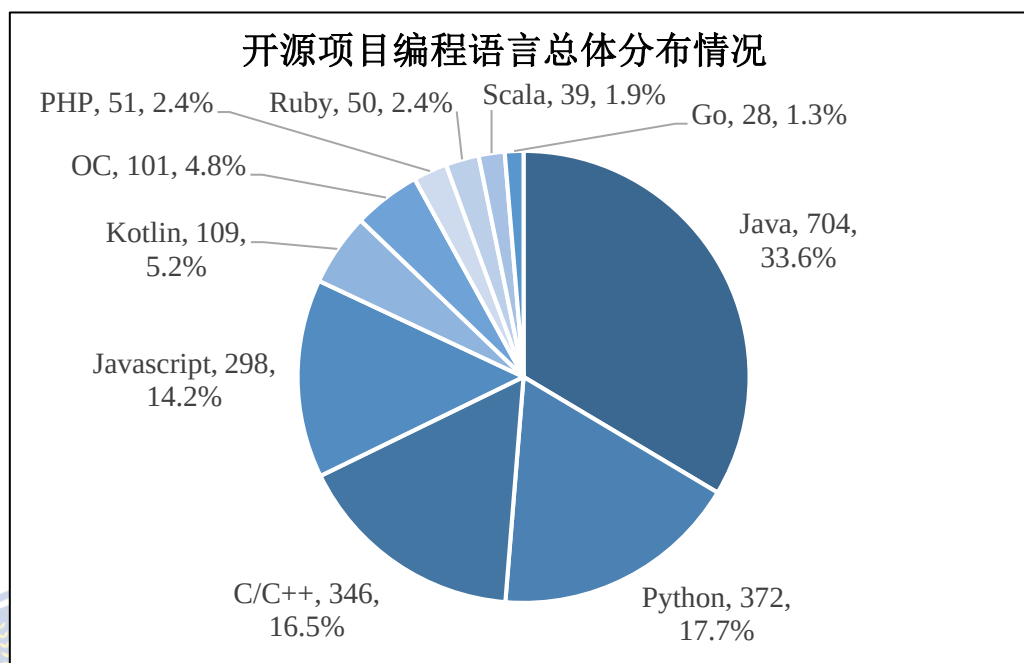
2、开源软件源代码安全状况分析

2022 年全年，“奇安信开源项目检测计划”对 2098 个开源软件项目的源代码进行了安全检测，代码总量为 173174712 行，共发现安全缺陷 3646486 个，其中高危缺陷 222640 个，整体缺陷密度为 21.06 个/千行，高危缺陷密度为 1.29 个/千行。三年来缺陷密度和高危缺陷密度均呈现出逐年上升的趋势。



(1) 编程语言分布情况

2022 年检测的 2098 个开源项目中，共涉及 10 种编程语言，分别是 Java、C/C++、Python、OC、Go、JavaScript、Kotlin、Scala、Ruby 和 PHP，编程语言的分布情况如下图所示。



(2) 典型安全缺陷检出情况

对 2098 个开源软件项目的缺陷检测结果分析发现，注入、密码管理、日志伪造、跨站脚本、NULL 引用、配置管理、输入验证、资源管理、路径遍历、API 误用等十类典型安全缺陷的总体检出率为 72.3%，与去年的 73.5%相当，但远高于前年的 56.3%。每类典型缺陷近三年的检出率及排名情况如下表所示。

排名	2022 检出率及排名	2021 检出率及排名	2020 检出率及排名
1	资源管理（35.0%）	路径遍历（36.7%）	输入验证（50.8%）
2	Null 引用（31.7%）	Null 引用（36.5%）	路径遍历（39.6%）

3	输入验证 (29.5%)	资源管理 (33.0%)	跨站脚本 (39.5%)
4	路径遍历 (28.6%)	输入验证 (25.1%)	注入 (37.3%)
5	注入 (21.8%)	密码管理 (23.5%)	Null 引用 (31.8%)
6	API 误用 (18.6%)	日志伪造 (22.9%)	资源管理 (31.6%)
7	日志伪造 (17.3%)	注入 (21.4%)	密码管理 (31.0%)
8	跨站脚本 (10.9%)	API 误用 (18.2%)	API 误用 (28.7%)
9	配置管理 (9.4%)	跨站脚本 (15.0%)	配置管理 (28.0%)
10	密码管理 (7.5%)	配置管理 (10.8%)	日志伪造 (18.2%)

可以看出,近三年典型缺陷的检出率和排名均有不同程度的变化。总体而言,除资源管理类缺陷小幅上升,输入验证、日志伪造类缺陷上下浮动外,其他7类缺陷的检出率均呈现出总体下降的趋势。

(3) 安全问题修复确认情况

2022年,“奇安信开源项目检测计划”共向各被测开源项目维护者反馈1484个安全问题(issues),其中547个得到确认并修复,其余的937个反馈不修复、未反馈,或无人处理(issues状态为Open),修复率仅为36.9%。开源项目维护者对安全问题的重视程度还较低。

通过对这些issues生命周期的分析发现,一个issue从提交到维护人员反馈确认并修复的时间,最快为一周,较长的可长达一年甚至更久。

3、开源软件公开报告漏洞状况分析

根据奇安信代码安全实验室监测与统计,截至2022年底,CVE/NVD、CNNVD、CNVD等公开漏洞库中共收录开源软件相关漏洞

57610 个，其中有 7682 个漏洞为 2022 年新增。

(1) 大型开源项目漏洞总数及年度增长 TOP20

截至 2022 年底，历史漏洞总数排名前 20 的大型开源项目信息如下表所示。Linux Kernel、Chromium (Google Chrome) 和 Mozilla Firefox 已连续三年排名前 3。

序号	大型开源项目	主页地址	历史漏洞总数
1	Linux Kernel	https://www.kernel.org/	5337
2	Chromium (Google Chrome)	http://www.chromium.org/	3072
3	Mozilla Firefox	https://www.mozilla.org/en-US/firefox/	2440
4	Thunderbird	https://www.thunderbird.net/zh-CN/	1427
5	MySQL	https://dev.mysql.com/	1346
6	Adobe Flash Player plugin	https://www.adobe.com/products/flashplayer/end-of-life.html	1089
7	linux-aws	https://aws.amazon.com/	855
8	GitLab	https://about.gitlab.com/	786
9	linux-gcp	https://launchpad.net/	770
10	linux-azure	https://azure.microsoft.com/	759
11	PHP	https://www.php.net/	740
12	SeaMonkey	https://www.seamonkey-project.org/	764
13	ImageMagick	https://imagemagick.org/index.php	642
14	Wireshark	https://www.wireshark.org/	635
15	WebKitGTK	http://webkitgtk.org/	649
16	Firefox ESR	https://www.mozilla.org/en-US/firefox/enterprise/	976
17	OpenJDK	https://openjdk.java.net/	585
18	Moodle	https://moodle.org/	495
19	Jenkins	https://www.jenkins.io/	483

20	Xen Project (Hypervisor)	https://xenproject.org/	486
----	--------------------------	---	-----

2022 年一年间，公开报告漏洞数量增长排名前 20 的大型开源项目信息如下表所示，Linux Kernel 依然是一年来增加漏洞最多的项目，达到 579 个。

序号	大型开源项目	主页地址	2022 年漏洞增量
1	Linux Kernel	https://www.kernel.org/	579
2	Chromium (Google Chrome)	http://www.chromium.org/	356
3	Mozilla Firefox	https://www.mozilla.org/en-US/firefox/	169
4	TensorFlow	https://www.tensorflow.org/	165
5	GitLab	https://about.gitlab.com/	161
6	MySQL	https://dev.mysql.com/	160
7	Thunderbird	https://www.thunderbird.net/zh-CN/	127
8	Jenkins	https://www.jenkins.io/	117
9	Vim	https://www.vim.org/	115
10	Firefox ESR	https://www.mozilla.org/en-US/firefox/enterprise/	106
11	gpac	http://gpac.io/	98
12	TeX Live	https://www.tug.org/texlive/	88
13	linux-aws	https://aws.amazon.com/	77
14	linux-azure	https://azure.microsoft.com/	74
15	linux-gcp	https://launchpad.net/	73
16	Microweber	https://microweber.org/	66
17	MariaDB	https://mariadb.org/	60
18	Xen Project (Hypervisor)	https://xenproject.org/	62
19	Liferay Portal	https://www.liferay.com/	49

20	Radare	https://www.radare.org/	48
----	--------	---	----

(2) 主流开源软件包生态系统漏洞总数及年度增长 TOP20

截至 2022 年底，主流开源软件包生态系统中历史漏洞总数排名前 20 的开源软件信息如下表所示。

序号	开源软件	所属包生态系统	历史漏洞总数
1	TensorFlow	Pypi	410
2	Chakra Core	Nuget	321
3	Jenkins	Maven	227
4	Electron - Cross-platform desktop application shell	NPM	196
5	Apache Tomcat	Maven	189
6	Electron - Cross-platform desktop application shell	Maven	183
7	TYPO3 CMS	Packagist	149
8	OpenSSL	Conan	147
9	FFmpeg-iOS	Swift	130
10	Magento Core	Packagist	119
11	Ruby on Rails	Rubygems	117
12	phpMyAdmin	Packagist	116
13	OpenSSL	Swift	105
14	Django	Pypi	98
15	Dolibarr ERP & CRM	Packagist	96
16	Drupal (core)	Packagist	93
17	Plone	Pypi	91
18	silverstripe-framework	Packagist	89
19	keycloak	Maven	85
20	Django	Conda	84

2022 年一年间，主流开源软件包生态系统中公开报告漏洞数量

增长排名前 20 的开源软件信息如下表所示。

序号	开源软件	所属包生态系统	2022 年漏洞增量
1	TensorFlow	Pypi	165
2	XWiki	Maven	44
3	Electron - Cross-platform desktop application shell	NPM	44
4	Electron - Cross-platform desktop application shell	Maven	41
5	rdiffWeb	Pypi	40
6	pimcore	Packagist	30
7	ms-mcms	Maven	30
8	Go programming language	Godoc	28
9	ShowDoc	Packagist	23
10	grafana	NPM	20
11	librenms	Packagist	20
12	CodeIgniter	Packagist	20
13	Mattermost	Godoc	20
14	GLPI	Packagist	19
15	Jenkins	Maven	19
16	incubator-airflow	Pypi	18
17	TYPO3 CMS	Packagist	18
18	libTIFF	Conan	17
19	Rocket.Chat	NPM	17
20	XWiki Platform	Maven	16

4、开源软件活跃度状况分析

我们依然把活跃度作为衡量开源软件安全性的一个维度。不活跃的开源软件，一旦出现安全漏洞，难以得到及时的修复；活跃的开源软件中，如果其版本更新发布的频率过高，同样会增加使用者运维的

成本和安全风险。

(1) 超 7 成开源软件项目处于不活跃状态

我们将超过一年未更新发布过版本的开源软件项目定义为不活跃项目。2022 年全年，主流开源软件包生态系统中不活跃的开源软件项目数量为 3967204 个，占比高达 72.1%，与去年的 69.9%和前年的 61.6%相比，呈现出逐年升高的趋势。

对八个典型的开源软件包生态系统进行分析和比较发现，除 Nuget 外，其他各包生态系统中不活跃项目的占比较去年报告数据均有小幅升高，Nuget 则从去年的 68.1%大幅下降至 54.3%。NPM 的不活跃项目数量最多，达 1693287 个，Rubygems 的不活跃项目比例仍为最高，占比高达 90.5%。具体数据见下表。

序号	包生态系统	项目总数	不活跃项目数	不活跃项目比例
1	Maven	656090	451557	68.8%
2	NPM	2328687	1693287	72.7%
3	Packagist	382623	283408	74.1%
4	Pypi	438973	289468	65.9%
5	Godoc	511587	348457	68.1%
6	Nuget	576800	313449	54.3%
7	Rubygems	173691	157230	90.5%
8	Swift	90136	78548	87.1%

(2) 超 2.2 万个开源软件一年内更新发布超过 100 个版本

2022 年全年，主流开源软件包生态系统中，更新发布 100 个以上版本的开源项目有 22403 个，明显高于去年的 19265 个和前年的

13411 个，与包生态系统中项目数量相同，呈现出逐年上升的趋势。八个典型的开源软件包生态系统中，一年内更新发布超过 100 个版本的项目数量见下表。

排名	包生态系统	对应的开发语言	一年内发布超过 100 个版本的项目数量
1	NPM	Javascript	12581
2	Maven	Java	3696
3	Nuget	.NET	2250
4	Godoc	Go	1572
5	Pypi	Python	989
6	Packagist	PHP	889
7	Rubygems	Ruby	248
8	Swift	Swift	18

5、关键基础开源软件分析

去年的报告中，我们新增了“关键基础开源软件”安全状况分析，即被众多其他开源软件所依赖（根据经验具体定义为直接依赖数大于 1000）的一类开源软件。今年的报告中延续了该部分内容，继续对这一类软件的安全性进行分析。

（1）主流开源生态关键基础开源软件 TOP50

去年报告中提到，从安全的视角来看，如果开源软件出现漏洞后，造成的放大效应越大，影响的范围越广，那么这个软件就越重要。本报告依然将直接依赖数大于 1000 的开源软件定义为关键基础开源软件。

分析发现,截止 2022 年底,Maven、NPM、Nuget、Pypi、Packagist、Rubygems 等主流开源生态中直接依赖数大于 1000 的开源软件共有 1254 款,较 2021 年底上涨 17.4%,直接依赖数排名 TOP50 的软件如下表。开源软件 junit:junit 的直接依赖数连续两年位居榜首,今年达 102980,较去年增加了 7366,也就是说一年之中新增 7366 款开源软件直接依赖于它。Apache Log4j2 依然排在 100 名以外。

排名	开源软件	所属包生态系统	直接依赖数
1	junit:junit	Maven	102980
2	rake	Rubygems	79723
3	org.scala-lang:scala-library	Maven	74610
4	bundler	Rubygems	69430
5	rspec	Rubygems	59289
6	Newtonsoft.Json	Nuget	56564
7	org.slf4j:slf4j-api	Maven	56378
8	tslib	NPM	54913
9	numpy	Pypi	46601
10	requests	Pypi	44962
11	chalk	NPM	41314
12	commander	NPM	39834
13	lodash	NPM	37988
14	illuminate/support	Packagist	33989
15	express	NPM	31928
16	guzzlehttp/guzzle	Packagist	31589
17	pandas	Pypi	30330
18	com.google.guava:guava	Maven	29254
19	org.jetbrains.kotlin:kotlin-stdlib-common	Maven	29165
20	axios	NPM	28386

21	pytest	Pypi	27525
22	ch.qos.logback:logback-classic	Maven	26770
23	inquirer	NPM	26691
24	org.mockito:mockito-core	Maven	25821
25	react	NPM	25690
26	request	NPM	25423
27	com.fasterxml.jackson.core:jackson-databind	Maven	23215
28	fs-extra	NPM	22381
29	org.apache.commons:commons-lang3	Maven	22168
30	org.jetbrains.kotlin:kotlin-stdlib-jdk8	Maven	22091
31	commons-io:commons-io	Maven	21855
32	matplotlib	Pypi	20418
33	scipy	Pypi	18959
34	org.clojure:clojure	Maven	18194
35	typescript	NPM	18112
36	org.projectlombok:lombok	Maven	17638
37	PyYAML	Pypi	16526
38	react-dom	NPM	16387
39	laravel/framework	Packagist	16287
40	click	Pypi	14788
41	moment	NPM	14756
42	log4j:log4j	Maven	14657
43	com.google.code.gson:gson	Maven	14624
44	pry	Rubygems	14548
45	org.slf4j:slf4j-log4j12	Maven	14051
46	org.assertj:assertj-core	Maven	13988
47	odoo	Pypi	13949
48	yiisoft/yii2	Packagist	13762

49	activesupport	Rubygems	13137
50	rails	Rubygems	13105

(2) 从未公开披露过漏洞的关键基础开源软件占比进一步升高

开源软件漏洞披露依然表现出参差不齐的状况，既有漏洞披露流程很正规的优秀开源项目，也有很多没有漏洞披露流程的开源项目。对 2022 年的数据分析发现，在 1254 款关键基础开源软件中，有 905 款从未公开披露过漏洞，占比为 72.2%，高于去年报告中的 66.6%。

造成这种现象的主要原因没变，依然有两个，即有的开源软件，特别是有的开源社区中的软件，漏洞虽然已被修复了，但没有记录和公开；另一方面是对一些开源软件安全性的关注度不够，对它们漏洞挖掘的研究还不多。

(3) 关键基础开源软件的整体运维风险仍处于较高水平

从“版本更新时间”和“周提交频率”两个维度来分析，依然有超 4 成的关键基础开源软件活跃度很低。其中，半年内没有发布过新版本的 key 基础开源软件有 520 款，占比为 41.5%，与去年数据相当；一年内周平均提交次数小于 5 和小于 1 的关键基础开源软件数量分别为 817 和 510，占比分别为 65.2%、40.7%，两项指标明显高于去年的 47.1%和 31.3%，说明关键基础开源软件提交的积极性有所降低。

在 TOP50 的关键基础开源软件中，只有 23 款软件明确已获得大厂或者基金会的支持，比去年还减少了一款，依然有 9 款软件的

Github 贡献者数量小于 100，NPM 生态中的 fs-extra 还是由 JP Richardson 个人来维护的。

整体来看，关键基础开源软件的运维安全状况未见改善，仍处于较高水平。

四、国内企业软件开发中开源软件应用状况

2022 年，奇安信代码安全实验室对 2631 个国内企业软件项目中使用开源软件的情况进行了分析，这些软件项目的应用领域涉及中小企业、金融、公共服务管理、通信、交通和能源等 6 个行业。

1、开源软件总体使用情况分析

(1) 平均每个软件项目使用 155 个开源软件

在被分析的 2631 个国内企业软件项目中，全部使用了开源软件，使用率为 100%。最多的项目使用了 5695 个开源软件，平均每个项目使用 155 个开源软件，较前两年的 127 个和 126 个有较大幅度的增加。使用开源软件最多的 10 个项目情况如下表所示。

项目名称	使用的开源软件数量
项目 1	5695
项目 2	4844
项目 3	4646
项目 4	4328
项目 5	3792
项目 6	3630
项目 7	3367

项目 8	2657
项目 9	2452
项目 10	2330

(2) 流行开源软件被超 4 成的软件项目使用

在分析的 2631 个国内企业软件项目中，使用最多的开源软件为 Apache Commons Lang，被 1146 个项目所使用，占比高达 43.6%，远高于去年的 36.2%和前年的 24.3%。被使用最多的前 10 名开源软件如下表所示。

开源软件名称	使用的项目数	被使用率
Apache Commons Lang	1146	43.6%
Simple Logging Facade for Java (SLF4J)	1135	43.1%
Commons IO	1035	39.3%
log4j	1026	39.0%
Apache Commons Codec	1017	38.7%
Jackson JSON processor	996	37.9%
Apache HttpClient (aka Apache HttpComponents Client)	988	37.6%
Spring Framework	986	37.5%
jackson-databind	976	37.1%
Apache Commons Collections	973	37.0%

2、开源软件漏洞风险分析

(1) 近 8 成软件项目存在容易利用的开源软件漏洞

分析发现，在 2631 个国内企业软件项目中，存在已知开源软件漏洞的项目有 2411 个，占比高达 91.6%；存在已知高危开源软件漏

洞的项目有 2268 个，占比为 86.2%；存在已知超危开源软件漏洞的项目有 2032 个，占比为 77.2%。这三个比例均比去年有所提高。

综合漏洞的 POC/EXP 情况以及 CVSS 可利用性指标等因素，我们将漏洞的利用难度分为容易、一般、困难，容易利用的漏洞风险极高。在分析的 2631 个项目中，存在容易利用的漏洞的项目有 2089 个，占比为 79.4%，高于去年的 77.0%。

(2) 平均每个软件项目存在 110 个已知开源软件漏洞

在 2631 个国内企业软件项目中，共检出 289066 个已知开源软件漏洞（涉及到 8289 个唯一 CVE 漏洞编号），平均每个软件项目存在 110 个已知开源软件漏洞，远高于前两年的 69 和 66 个。最多的软件项目存在 1566 个已知开源软件漏洞。存在已知开源软件漏洞数量排名前 10 的项目情况如下表所示。

项目	存在开源软件漏洞数量
项目 1	1566
项目 2	1506
项目 3	1459
项目 4	1411
项目 5	1407
项目 6	1395
项目 7	1388
项目 8	1234
项目 9	1169
项目 10	1073

(3) 影响最广的开源软件漏洞存在于超 4 成的软件项目中

从漏洞的影响度来分析，影响范围最大的开源软件漏洞为 CVE-2022-42003 和 CVE-2022-42004，存在于 41.9% 的软件项目中，比去年最高的 31.7% 高出约 10%，足以说明它们影响广泛。2022 年影响度排名前 10 的开源软件漏洞如下表所示。

漏洞名称	CVE 编号	影响项目数量	影响度
FasterXML jackson-databind 代码问题漏洞	CVE-2022-42003	1103	41.9%
FasterXML jackson-databind 代码问题漏洞	CVE-2022-42004	1102	41.9%
Spring Framework 安全漏洞	CVE-2023-20861	1078	41.0%
Spring Framework 安全漏洞	CVE-2023-20863	1077	40.9%
Apache Commons FileUpload 安全漏洞	CVE-2023-24998	1012	38.5%
Spring Framework 远程代码执行漏洞	CVE-2022-22965	1006	38.2%
Spring Framework 输入验证错误漏洞	CVE-2022-22970	994	37.8%
VMware Spring Framework 代码问题漏洞	CVE-2016-1000027	979	37.2%
FasterXML jackson-databind 代码问题漏洞	CVE-2023-35116	976	37.1%
VMware Spring Framework 安全特征问题漏洞	CVE-2022-22968	970	36.9%

容易利用的漏洞更易被用来发起攻击，风险极高。影响度排名前 10 的容易利用的开源软件漏洞情况如下表所示。

容易利用的漏洞名称	CVE 编号	影响项目数量	影响度
Spring Framework 远程代码执行漏洞	CVE-2022-22965	1006	38.2%
VMware Spring Framework 代码问题漏洞	CVE-2016-1000027	979	37.2%
VMware Spring Framework 安全漏洞	CVE-2021-22060	896	34.1%

Vmware Spring Framework 安全漏洞	CVE-2021-22096	886	33.7%
Apache HttpClient 安全漏洞	CVE-2020-13956	869	33.0%
jQuery 跨站脚本漏洞	CVE-2020-11022	806	30.6%
jQuery 跨站脚本漏洞	CVE-2020-11023	806	30.6%
jQuery 跨站脚本漏洞	CVE-2019-11358	778	29.6%
Fastjson 代码问题漏洞	CVE-2022-25845	747	28.4%
FasterXML jackson-databind 代码问题漏洞	CVE-2020-8840	737	28.0%

(4) 20 多年前的开源软件漏洞仍然存在于多个软件项目中

分析发现，与前两年报告结果一致，部分软件项目中仍然存在很久之前公开的古老开源软件漏洞，其中，最古老的漏洞是 2002 年 3 月 15 日公开的 CVE-2002-0059，距今已 21 年，仍然存在于 11 个项目中。部分古老开源软件漏洞的影响情况如下表所示。

漏洞名称	CVE 编号	发布日期	影响项目数量
zlib 安全漏洞	CVE-2002-0059	2002-03-15	11
LibPNG 超阔页边空白图象处理内存破坏漏洞	CVE-2002-0660	2002-06-19	1
Portable Network Graphics 任意脚本远程执行漏洞	CVE-2002-1363	2002-12-26	4
Linux Kernel 混杂模式状态漏洞	CVE-2002-1976	2002-12-31	1
zlib 安全漏洞	CVE-2003-0107	2003-02-23	17
TCP 协议安全漏洞	CVE-2004-0230	2004-08-18	1
LibPNG 不合法 PNG 越界访问拒绝服务漏洞	CVE-2004-0421	2004-08-18	1
Microsoft MSN Messenger PNG 图片解析远程代码执行漏洞 (MS05-009)	CVE-2004-0597	2004-11-23	4
libpng png_handle_iccp NULL 指针远程拒绝服务漏洞	CVE-2004-0598	2004-11-23	4
libpng 逐行读取远程整数溢出漏洞	CVE-2004-0599	2004-11-23	4

3、开源软件许可协议风险分析

(1) 最流行的开源许可协议在超半数的项目中使用

在 2631 个被分析的国内企业软件项目中，共发现 537 个开源许可协议。涉及项目数最多的协议依然是 Apache License 2.0，在 1349 个项目中使用，占比为 51.3%，虽然超过半数，但低于去年的 77.1%。涉及软件项目数排名前 10 的许可协议如下表所示。

开源许可协议	涉及项目数	所占比例
Apache License 2.0	1349	51.3%
MIT License	1050	39.9%
BSD 3-Clause "New" or "Revised" License	328	12.5%
GNU General Public License v2.0 only	157	6.0%
GNU General Public License v1.0 or later	147	5.6%
GNU General Public License v2.0 or later	123	4.7%
Public Domain	122	4.6%
BSD 2-Clause "Simplified" License	113	4.3%
GNU Lesser General Public License v2.1 or later	91	3.5%
GNU General Public License v3.0 only	86	3.3%

(2) 1/6 的项目使用了含有超、高危许可协议的开源软件

有些类型的开源许可协议中包含了一些限制性条款，在使用过程中一旦违反这些条款，可能对企业的商业利益和声誉造成严重的损害。这些限制性条款包括如：“禁止本软件及其衍生品用于商业目的”、“禁止在未支付费用和版税的情况下将该软件用于非公开、非商业用途”、“软件发布时必须公开软件的源代码”等。

奇安信代码安全实验室将限制性较为苛刻的一类协议定义为超危许可协议，将限制性较大而未达到超危水平的一类协议定义为高危许可协议。在 2631 个被分析的国内企业软件项目中，存在超危许可协议的项目 142 个，占比 5.4%；存在高危许可协议的项目 301 个，占比 11.4%。含超危和高危许可协议的项目合计占比 16.8%，约 1/6。

涉及软件项目数排名前 10 的超危许可协议如下表所示。

超危开源许可协议	涉及项目数
GNU General Public License v3.0 only	86
GNU General Public License v3.0 or later	82
GNU Affero General Public License v3.0 only	57
GNU Lesser General Public License v3.0 only	46
GNU Lesser General Public License v3.0 or later	43
GNU Free Documentation License v1.2	18
GNU Free Documentation License v1.2 only	7
GNU General Public License v3.0 or later	7
GNU General Public License v3.0 only	2
Creative Commons Attribution Non Commercial Share Alike 2.5 Generic	1

涉及软件项目数排名前 10 的高危许可协议如下表所示。

高危开源许可协议	涉及项目数
GNU General Public License v2.0 only	157
GNU General Public License v2.0 or later	123
GNU Lesser General Public License v2.1 or later	91
GNU Library General Public License v2 or later	78
GNU Lesser General Public License v2.1 only	72
Mozilla Public License 1.1	48
Eclipse Public License 1.0	45

Creative Commons Attribution Share Alike 3.0 Unported	41
GNU Library General Public License v2 only	31
GNU Lesser General Public License v2.1 only	30

4、开源软件运维风险分析

开源软件运维风险复杂多样，报告主要从老旧开源软件的使用和开源软件多版本的使用角度进行分析。

(1) 近 30 年前的老旧开源软件版本仍在被使用

分析发现，与前两年报告数据一致，许多软件项目中仍然在使用老旧的开源软件版本，有的甚至是近 30 年前发布的版本，存在很大的运维风险。被使用的老旧开源软件版本中，最早的一款是 1995 年 8 月 2 日发布的 IJG JPEG 6，已经有 28 年之久，但仍然被软件项目所使用。按老旧程度排名前 10 的开源软件如下表所示。

开源软件名称	版本号	版本发布日期	使用它的项目数量
IJG JPEG	6	1995-08-02	1
IJG JPEG	6a	1996-02-07	13
glut	3.1-3	1996-12-11	1
zlib	1.1.3	2001-02-05	34
libpng	1.0.12	2001-06-09	3
libpng	1.2.1	2001-12-15	8
Saxpath	1.0-FCS	2002-05-14	1
jaxen-core from jdom 1.1 distribution	1.0-FCS	2002-05-14	1
libpng	1.2.3	2002-05-24	4
Apache Xalan	2.5.D1	2003-03-03	5

(2) 开源软件各版本使用依然混乱

分析发现，各个项目中开源软件使用的版本依然非常混乱，并非使用的都是最新版本。Spring Framework 是被使用版本最多的开源软件，有 181 个版本在被使用，比去年的 235 低，但高于前年的 162，说明开源软件版本使用混乱的状况依然严重。按照被使用版本的数量排序，排名前 10 的开源软件情况如下表所示。

开源软件名称	被使用的版本数量
Spring Framework	181
@types/node	132
Apache Tomcat	130
Spring Boot	126
Netty Project	118
Hibernate ORM	117
jackson-databind	112
Spring TestContext Framework	109
Jackson JSON processor	109
Jetty	105

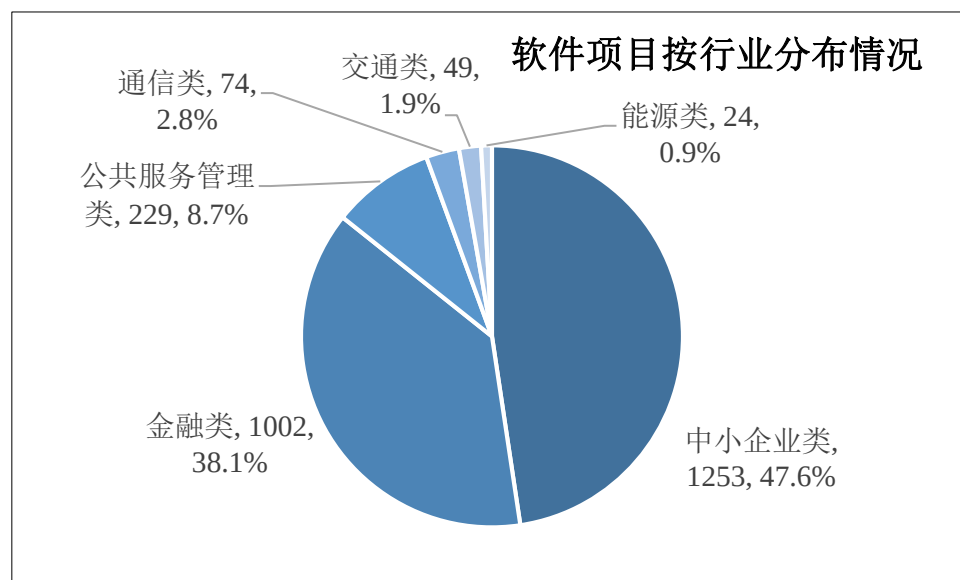
5、不同行业软件项目开源使用风险分析

所有 2631 个被分析的国内企业软件项目，按照应用领域、功能属性，可划归六个行业，即中小企业类、金融类、公共服务管理类、通信类、交通类和能源类软件项目。

(1) 各行业软件项目分布情况

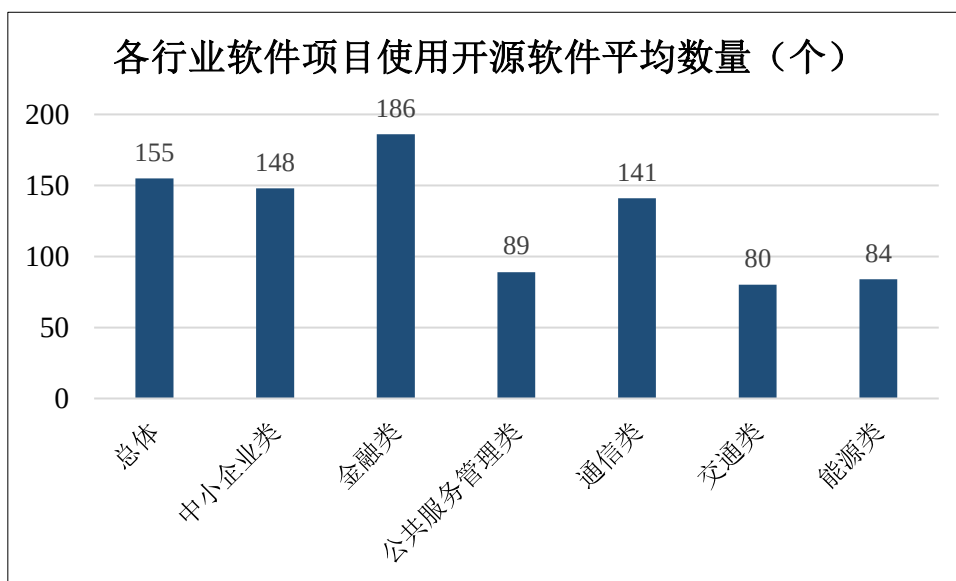
在 2631 个国内企业自主开发的软件项目中，数量排名前 3 的依

次为中小型企业开发的软件、金融类项目和公共服务管理类项目，项目数分别为 1253 个、1002 个和 229 个，它们合计占比接近 95%。被分析软件项目按行业的分布情况如下图所示。



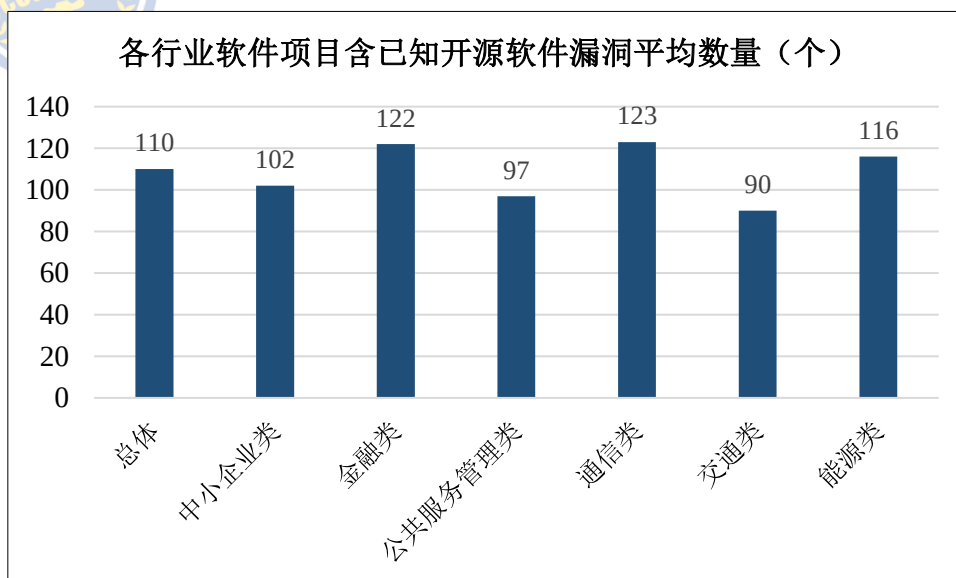
(2) 各行业软件项目使用开源软件情况

各行业软件项目使用开源软件的平均数量如下图所示。与总体平均数 155 个相比，中小企业类、通信类项目的开源软件使用平均数与之相当；公共服务管理类、交通类、能源类项目使用开源软件较少，平均数在 80-89 个之间；金融类项目使用开源软件平均数量最大，且远高于其他类别和总体水平，达到 186 个。



（3）各行业软件项目含已知开源软件漏洞情况

各行业软件项目含已知开源软件漏洞的平均数量如下图所示。可以看出，通信类、金融类、能源类软件项目的已知开源软件漏洞平均数高于总体，前两者已高于 120 个；其他行业的软件项目虽然平均漏洞数较低，但也明显高于去年和前年的总体平均水平(69 个和 66 个)。



五、典型软件供应链安全风险实例分析

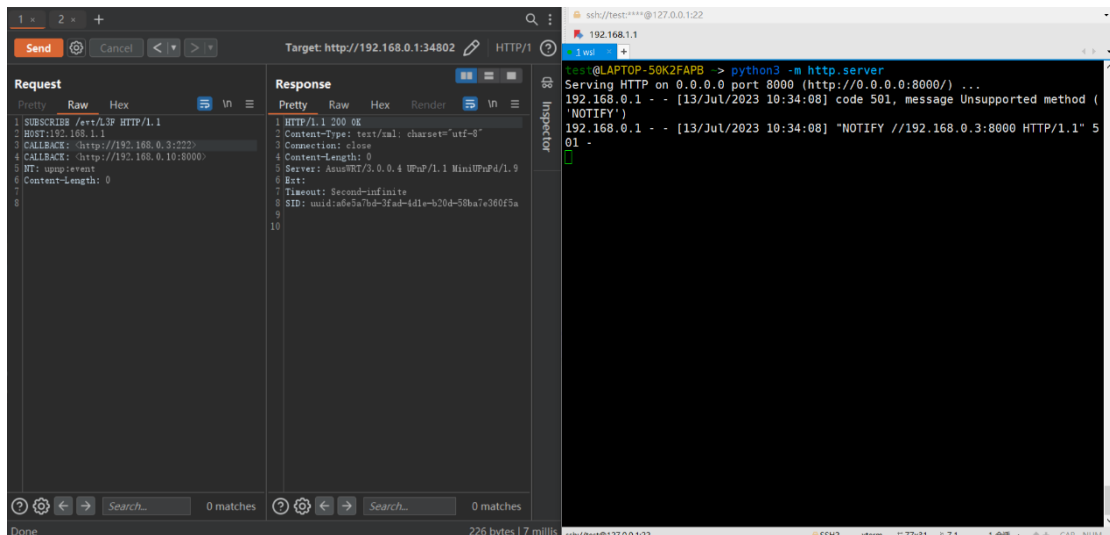
1、某主流企业级无线路由器供应链攻击实例分析

某国际知名厂商生产的企业级千兆无线路由器，广泛应用于各类企业中，对该路由器的固件进行分析发现，固件使用了开源工具 MiniUPnPd 1.9。

MiniUPnPd 是一款可用于嵌入式系统的通用即插即用开源工具，它是物联网核心协议 UPnP 的一个实现。UPnP 协议允许各种网络设备在没有特殊设置或配置的情况下进行通信，使得设备彼此间可自动连接和协同工作。例如，新打印机通电并接入网络后，局域网中的计算机自动获取打印机的型号等信息，方便进行驱动安装。

CVE-2020-12695 是 UPnP 协议的一个高危历史漏洞，又被称作 CallStranger 漏洞，攻击者可利用它绕过内网的数据防泄漏（DLP）系统，实现数据逃逸，从而导致敏感数据泄露；还可以对设备所在内网进行扫描，甚至还能劫持设备进行分布式拒绝服务（DDoS）攻击。该漏洞影响产品的范围非常广泛。

下图展示了在本例的路由器上利用 CVE-2020-12695 的效果，可以扫描路由器所在内网中某机器的 222 端口是否开启，若未收到连接请求，则确定端口开启，进而为后续网络攻击提供基础。



本实例中，软件供应链风险传播链条如下：某主流无线路由器的固件使用开源工具 MiniUPnPd 来实现 UPnP 服务，当 UPnP 服务功能开启时，UPnP 协议漏洞会影响该路由器，对路由器所在内网进行扫描，由此引发软件供应链攻击。

2、ZCS 协同办公系统供应链攻击实例分析

Zimbra Collaboration Suite (ZCS) 是一款开源协同办公套件，包括邮件服务、日历、通信录等功能。ZCS 邮件服务器目前应用于全球 140 个国家的 20 多万个组织中，包括 1000 多个政府和金融机构。ZCS 的防病毒引擎 Amavis 对电子邮件中的附件提取时使用了开源组件 cpio（一款 GNU/Linux 实用程序，旨在提取各种存档格式，包括.cpio、.tar 和.rpm 等）。

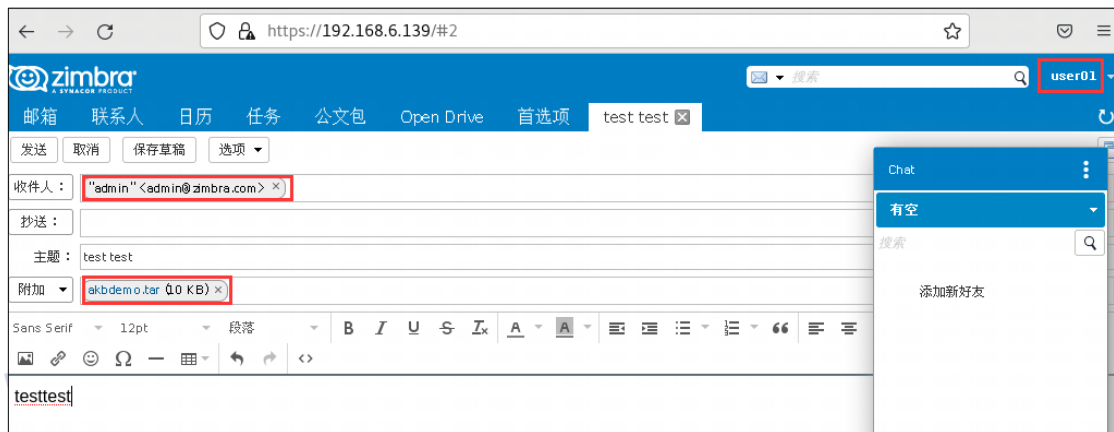
CVE-2022-41352 是影响 ZCS 8.8.15 和 9.0 的一个超危历史漏洞，因 Amavis 使用 cpio 提取存档文件，而 cpio 又存在可在提取的存档中嵌入绝对路径的目录遍历问题，这使得未经身份验证的远程攻击者可通过 Amavis 将任意文件写入用户可访问的文件系统的任何路

径，从而实现在 Web 根目录中植入 shell，最终导致在目标系统上任意代码执行。攻击者已利用该漏洞攻陷了数千台服务器。

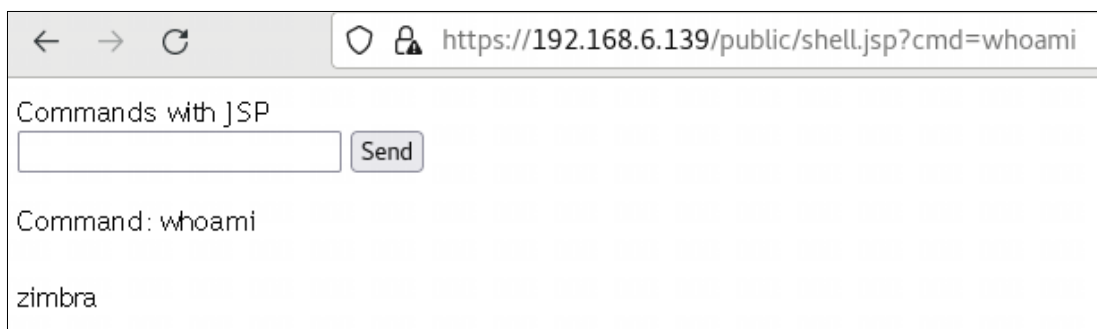
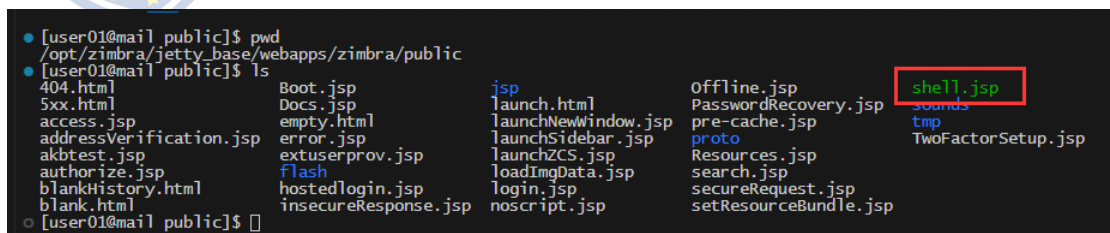
我们分析验证发现，目前依然可以利用该历史漏洞，通过 ZCS 8.8.15 patch34 和 9.0.0 patch27 在系统上实现任意文件上传。下面的一组图片展示了验证过程。

- 实现任意文件上传的效果

- 上传任意文件



- 成功上传恶意文件到指定目录



- 验证是 cpio 的历史漏洞导致了任意文件上传

- 手动下载恶意 tar 文件



- 使用 cpio 提取 tar 文件后成功上传恶意文件到指定目录

```
[root@mail 下载] # ls /opt/zimbra/jetty_base/webapps/zimbra/public/
404.html      blank.html    flash         launchSidebar.jsp PasswordRecovery.jsp setResourceBundle.jsp
5xx.html      Boot.jsp      hostedlogin.jsp launchZCS.jsp    pre-cache.jsp      shell.jsp
access.jsp     Docs.jsp      insecureResponse.jsp loadImgData.jsp  proto              sounds
addressVerification.jsp empty.html    jsp           login.jsp        Resources.jsp       tmp
authorize.jsp  error.jsp     launch.html   noscript.jsp     search.jsp          TwoFactorSetup.jsp
blankHistory.html extuserprov.jsp launchNewWindow.jsp Offline.jsp       secureRequest.jsp

[root@mail 下载] # ls
akbdemo.tar
[root@mail 下载] # sudo cpio --no-absolute-filenames -ivf akbdemo.tar
akbdemo
akbdemo/akbtest.jsp
4 块
[root@mail 下载] # ls
akbdemo akbdemo.tar
[root@mail 下载] # ls /opt/zimbra/jetty_base/webapps/zimbra/public/
404.html      Boot.jsp      jsp           Offline.jsp      shell.jsp
5xx.html      Docs.jsp      launch.html   PasswordRecovery.jsp sounds
access.jsp     empty.html    launchNewWindow.jsp pre-cache.jsp    tmp
addressVerification.jsp error.jsp     launchSidebar.jsp proto            TwoFactorSetup.jsp
akbtest.jsp   extuserprov.jsp launchZCS.jsp  Resources.jsp
authorize.jsp  flash        loadImgData.jsp search.jsp
blankHistory.html hostedlogin.jsp login.jsp      secureRequest.jsp
blank.html    insecureResponse.jsp noscript.jsp  setResourceBundle.jsp

[root@mail 下载] #
```

Browser address bar: <https://192.168.6.139/public/akbtest.jsp>

Page content: Hello world!

本实例中，软件供应链风险传播链条如下：ZCS 的防病毒引擎 Amavis 使用了开源组件 cpio 来提取存档格式，cpio 的历史漏洞影响到了 ZCS，可在安装有 ZCS 的系统上实现任意文件上传的软件供应链攻击。

3、多款国产操作系统供应链攻击实例分析

CLS_FLOWER 是 Linux 内核网络子系统中的一个分类器模块，它允许基于流来控制数据包，这些流通过匹配已知的数据包字段和元数

据来确定。CLS_FLOWER 及相关模块的卸载为基于流的系统的用户提供了一种强大的机制,可以增加吞吐量并降低 CPU 利用率。分析发现,CLS_FLOWER 广泛应用于基于 Linux 内核的国产操作系统中,其历史漏洞可能引发针对这些操作系统的软件供应链攻击。验证发现,至少 4 款国产操作系统会受到 CLS_FLOWER 历史漏洞的影响。

以某主流国产操作系统为例,它是一款以桌面应用为主的开源操作系统,包含数十款应用软件,支持笔记本、台式机等。经分析发现,该国产操作系统的 2023 年 6 月版本基于 Linux Kernel 5.15.x,也使用了 CLS_FLOWER 模块。

CVE-2023-35788 是 CLS_FLOWER 的一个高危历史漏洞,影响 Linux Kernel 6.3.7 之前版本,攻击者可利用此漏洞字节越界写的特性绕过参数校验函数,从而实现普通用户本地权限提升,危害很大。

下图展示了在该国产操作系统上利用 CVE-2023-35788 的效果,可以看到,成功攻击后完成了从普通用户 user 到超级用户 root 的切换,实现了权限提升。

```
user@linux:~/Documents/poc$ id
uid=1000(user) gid=1000(user) groups=1000(user),7(lp),27(sudo),100(users),109(netdev),118(lpadmin),119(scanner),120(sambashare)
user@linux:~/Documents/poc$ ./run.sh
Saved state !
argv[0]: ./poc
socket_fd:5
fuse create success! evil_fd:6
recv num: 1100
recv num 2: 20
leak func addr: 0xffffffffb87aa960
leak addr base: 0xffffffffb820000
recv num: 1192
recv num 2: 20
chain 667 delete
heap spray 1 over
leak slab 1 addr 0xffff8eb075a96200
chain 666 delete
modprobe_path have been changed!
root@linux:~/Documents/poc# id
uid=0(root) gid=0(root) groups=0(root),7(lp),27(sudo),100(users),109(netdev),118(lpadmin),119(scanner),120(sambashare),1000(user)
root@linux:~/Documents/poc# whoami
root
```

本实例中，软件供应链风险传播链条如下：国产操作系统基于 Linux 内核开发，使用了 Linux 内核网络子系统中的 CLS_FLOWER 模块，攻击者可利用此模块的历史漏洞对这些国产操作系统实施权限提升方面的软件供应链攻击。

4、某国产 CMS 系统供应链攻击实例分析

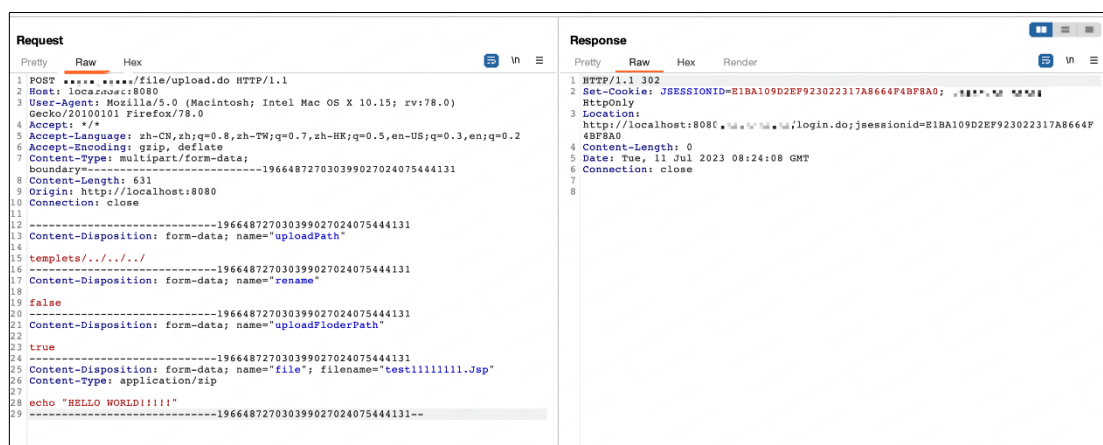
某国产 CMS 系统用于管理和发布网站内容，操作方便，已在国内多家行政单位、科研院校、企业、大型商业银行中使用。分析发现，该国产 CMS 系统基于 Spring Framework 框架开发，使用 Apache Shiro 1.4.0 进行安全权限管理。

Apache Shiro 是一款功能强大且易用的安全开源框架，具有身份验证、授权、密码和会话管理等功能，可以支持常用的 Spring Framework 框架，在国内外各类 Java Web 服务中被广泛应用。

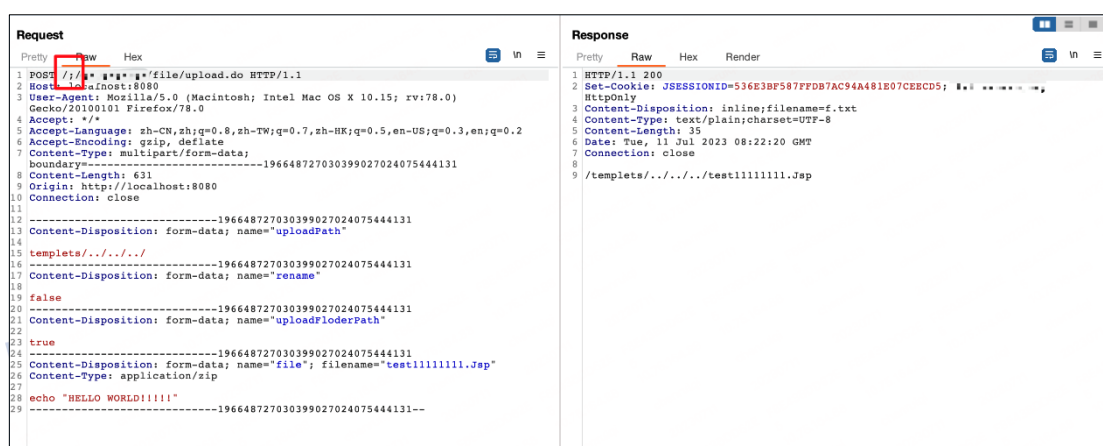
CVE-2020-1957 是 Apache Shiro 的一个高危历史漏洞，影响 Shiro 1.5.2 之前的版本。利用该漏洞可实现身份认证绕过，结合其他漏洞可实现任意文件上传等攻击效果，危害巨大。

下面几张图片展示了我们利用该漏洞实现无需身份认证而操作后台，以及结合使用其他后台漏洞实现任意文件上传的验证过程。

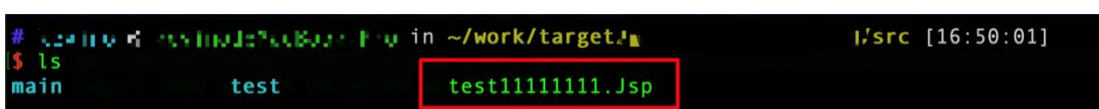
- 未授权访问后台接口，响应跳转到登陆页面



- 使用 CVE-2020-1957 绕过 Shiro 权限检查，未经授权而调用 CMS 后台文件上传接口



- 成功上传恶意文件到指定目录



本实例中，软件供应链风险传播链条如下：Apache Shiro 被某国产 CMS 系统所使用，其高危历史漏洞可影响使用该 CMS 系统的后台，导致可针对其实施权限绕过，并结合其他漏洞实施任意文件上传等软件供应链攻击。

六、总结及建议

过去的一年里，开源软件供应链安全成为关注的焦点，被推到了一个前所未有的高度。开源软件供应链因具有开源软件使用广泛、依赖关系普遍且复杂、开源许可体系繁杂、开源托管平台很大程度上依赖于国外等特点，其安全问题更加突出和显著。

近年来，欧美等一些国家和地区从政府、基金会、科技巨头企业等多个层面推进和出台了一系列针对开源治理的措施和解决方案；国内在这方面也在不断发力：在多个法规、规划中都提到了促进开源社区发展和开源治理等内容；国家标准《软件产品开源代码安全评价方法》和《软件供应链安全要求》也在持续推进中，将为开源软件的安全评价和供应链治理提供顶层指导；一些规模较大的机构和企业也已建立起了针对开源软件的管理流程和方法。

但另一方面我们也看到，国内大多数机构和企业目前对于开源软件供应链的管理还没有一致的规范，流程、方法千差万别，主要体现在缺少统一的开源治理体系框架，缺少被普遍认可的开源治理指南，缺少统一的开源治理公共服务平台的支撑等方面。

对于开源软件，使用者应做到使用前的安全评估，即借助软件成分分析和代码审计等自动化工具对新引入的开源组件进行安全测试、漏洞分析和综合评估等工作；使用中的积极维护，即建立安全开源组件仓库，及时更新版本，主动进行安全测试；使用后的安全响应，即建立开源软件漏洞的多方协作响应和处理机制，提升漏洞情报预警和

通报能力。具体建议如下：

1) 制定具有普遍共识的开源治理框架和指南

加快编制和出台开源供应链安全相关的顶层指导意见和国标，形成开源治理总体框架体系，包括《软件产品开源代码安全评价方法》、《软件供应链安全要求》、软件物料清单（SBOM）相关的标准等；基于开源治理相关标准细化实施方法，包括具体技术、推荐工具、机制方法等内容，形成具体指南，提升开源软件供应链安全保护工作的可操作性和效率。可在指南中推荐相关的自动化工具，并将国产优秀工具作为首选，以增强指南的落地性和实操性。

2) 建立开源治理平台和漏洞响应机制，方便各方开展开源软件资产风险监测

建立国家级/行业级开源软件供应链风险治理平台，具备系统化、规模化的软件源代码缺陷和后门分析、软件漏洞分析、开源软件成分及风险分析、SBOM 检测验证等能力，能够及时发现和处置开源软件供应链安全风险；建立完善的开源软件漏洞响应渠道和机制，包括开源软件漏洞信息的收集、漏洞报告渠道的建立和维护、漏洞补丁的通知和分发、漏洞应急响应和修复支持等；软件产品使用方和开发方均应严控上游，尤其重点管控开源软件的使用，开源软件各相关方应建立开源软件资产台账，建议采用可融入软件开发流程的开源安全治理工具和平台，持续监测和消减所使用开源软件的安全风险。

3) 提升针对开源软件供应链的“五防”能力

针对开源软件供应链，应提升“五防”能力，即防漏洞、防投毒、

防侵权、防停服、防断供。提升“五防”能力的关键有两点，一是提升软件分析的准确性和效率，二是提升软件透明度，尽量获取软件，特别是第三方、开源软件的基本信息，包括安全漏洞、恶意代码、许可协议、来源等方面的信息。建议以 SBOM 作为基础，提升“五防”能力：软件厂商和供应商应采用工具监测所使用的开源软件物料，应提取、生成并向客户提供产品 SBOM，持续监测软件产品物料的安全漏洞等风险情况；软件用户应要求厂商和供应商提供 SBOM，验证其正确性，并基于 SBOM 持续跟踪相关的威胁情报。



奇安信代码安全实验室
— QI-ANXIN Codesafe Team —

附录：奇安信代码安全实验室简介

奇安信代码安全实验室是奇安信集团旗下，专注于软件源代码安全分析技术、二进制漏洞挖掘技术与开发的团队。实验室支撑国家级漏洞平台的技术工作，多次向国家信息安全漏洞库（CNNVD）和国家信息安全漏洞共享平台（CNVD）报送原创通用型漏洞信息并获得表彰；帮助微软、谷歌、苹果、Cisco、Juniper、Red Hat、Ubuntu、Oracle、Adobe、VMware、阿里云、飞塔、华为、施耐德、Mikrotik、Netgear、D-Link、Netis、ThinkPHP、以太坊、Facebook、亚马逊、IBM、SAP、Netflix、Kubernetes、Apache 基金会、腾讯、滴滴等大型厂商和机构的商用产品或开源项目发现了数百个安全缺陷和漏洞，并获得公开致谢。目前，实验室拥有国家信息安全漏洞库（CNNVD）特聘专家一名，多名成员入选微软全球 TOP 安全研究者、Oracle 安全纵深防御计划贡献者等精英榜单。在 Pwn2Own 2017 世界黑客大赛上，实验室成员还曾获得 Master of Pwn 破解大师冠军称号。

基于奇安信代码安全实验室多年的技术积累，奇安信集团在国内率先推出了自主可控的软件代码安全分析系统——奇安信代码卫士和奇安信开源卫士。奇安信代码卫士是一套静态应用程序安全测试系统，可检测 2600 多种源代码安全缺陷，支持 C、C++、C#、Objective-C、Swift、Java、JavaScript、PHP、Python、Cobol、Go 等 20 多种编程语言。奇安信开源卫士是一套集开源软件识别与安全管控于一体的软件成分风险分析系统，通过智能化数据收集引擎在全球范围内广

泛获取开源软件信息和漏洞信息，帮助客户掌握开源软件资产状况，及时获取开源软件漏洞情报，降低由开源软件带来的安全风险，奇安信开源卫士目前可识别超过 1.3 亿个开源软件版本，兼容 NVD、CNNVD、CNVD 等多个漏洞库。奇安信代码卫士和奇安信开源卫士目前已经在数百家大型企业和机构中应用，帮助客户构建自身的代码安全保障体系，消减软件代码安全隐患，并入选国家发改委数字化转型伙伴行动、工信部中小企业数字化赋能专项行动，为中小企业提供软件代码安全检测平台和服务。



奇安信代码安全实验室
— QI-ANXIN Codesafe Team —